

Ocenę celującą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- pisze program rozkładający liczbę złożoną na dwie liczby pierwsze (hipoteza Goldbacha),
- implementuje w języku C++ algorytm Euklidesa, stosując iterację.
- bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,
- pisze programy o podwyższonym stopniu trudności
- optymalizuje rozwiązania,
- stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL),
- dobiera struktury danych i metody do rodzaju problemu,
- wykonuje operacje arytmetyczne na liczbach w różnych systemach,
- wykorzystuje algorytm Euklidesa do działań na ułamkach,
- pisze program wyszukujący jednocześnie minimum i maksimum w zbiorze z wykorzystaniem metody „dziel i zwyciężaj” oraz podaje wzór na liczbę wykonywanych operacji,
- wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach,
- pisze programy wyszukujące lidera i idola w zbiorze, optymalizuje je, szacuje złożoność czasową,
- do implementacji rozszerzonego algorytmu Euklidesa stosuje iterację,
- stosuje metody zachłanną i dynamiczną w problemach kasjera, harmonogramu wykorzystania sali, pakowania plecaka i wyszukiwania drogi, wskazuje wady i zalety obu metod, szacuje złożoność czasową,
- aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy.

Ocenę dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- określa specyfikację algorytmu (dane, wynik),
- pisze programy o różnym stopniu trudności, szacuje ich efektywność,
- przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, schematu blokowego, pseudokodu,
- dobiera typy danych do realizacji problemu,
- stosuje zmienne typu unsigned w tworzonych programach,
- implementuje w języku C++ i optymalizuje algorytm sprawdzający, czy liczba jest pierwsza,
- pisze program rozkładający liczby na czynniki pierwsze,
- stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW,
- wykorzystuje algorytm Euklidesa do działań na ułamkach,
- stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową,
- stosuje w programach algorytmy wyszukiwania lidera i idola w zbiorze,
- zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie,
- stosuje metodę zachłanną w programach – problem kasjera, harmonogram wykorzystania sali, wyszukiwanie drogi, pakowanie plecaka,

- uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej.

Ocenę dostateczną otrzymuje uczeń, który spełnia kryteria oceny dopuszczającej, a ponadto:

- przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu, schematu blokowego,
- dodaje liczby binarne,
- konwertuje liczby między pozycyjnymi systemami liczbowymi,
- wykonuje działania arytmetyczne na liczbach w systemach liczbowych o różnych podstawach,
- przedstawia liczby w kodzie U2,
- definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne (koniunkcja, alternatywa, negacja) oraz operatory logiczne,
- charakteryzuje wybrane typy zmiennych służących do zapisu liczb całkowitych w języku C++: short int, int, long int, long long int,
- pisze programy wykonujące działania na liczbach całkowitych,
- wyjaśnia różnicę między parametrami formalnym i aktualnym, a także między zmiennymi lokalną i globalną,
- implementuje w języku C++ algorytm naiwny sprawdzający, czy liczba jest pierwsza,
- implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem,
- implementuje w języku C++ algorytmy wyszukiwania liniowego i liniowego z wartownikiem, porównuje ich efektywność,
- omawia algorytm znajdowania idola i lidera w zbiorze,
- implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi,
- omawia rozszerzony algorytm Euklidesa,
- formułuje algorytm wydawania reszty minimalną liczbą monet, harmonogramu wykorzystania sali, pakowania plecaka, znajdowania drogi metodami zachłanną i dynamiczną,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.

Ocenę dopuszczającą otrzymuje uczeń, który:

- definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja,
- wymienia sposoby reprezentacji algorytmów,
- korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki,
- pisze programy o niewielkim stopniu trudności,
- omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia),
- korzysta z podstawowych funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych,
- wymienia podstawowe typy danych, operacje arytmetyczne i logiczne,
- w pisanych programach korzysta ze strukturalnych typów danych: tablic,
- definiuje pojęcie systemów liczbowych,
- wymienia systemy liczbowe używane w informatyce,

- konwertuje liczby między systemami binarnym i decymalnym,
- dodaje pisemnie liczby binarne,
- podaje definicje liczby pierwszej i liczby złożonej,
- implementuje w języku C++ algorytm zliczający dzielniki danej liczby,
- omawia geometryczną interpretację algorytmu Euklidesa,
- wyjaśnia, na czym polega metoda „dziel i zwyciężaj”,
- omawia algorytmy wyszukiwania liczby w zbiorach uporządkowanym i nieuporządkowanym,
- stosuje funkcję losującą w tworzonych programach,
- wypisuje liczby pierwsze z zadanego przedziału, stosując metodę sita Eratostenesa,
- wyszukuje w ciągu liczb spójne podciągi (nierosnący, niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę,
- wskazuje idola i lidera w zbiorze danych,
- definiuje pojęcia iteracji,
- omawia zasadę złotego podziału,
- opisuje rozszerzony algorytm Euklidesa,
- omawia metody zachłanne na przykładzie problemu kasjera, harmonogramu sali, pakowania plecaka i wyszukiwania drogi,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności.